

Introdução ao Linux

Aula VIII: Shell Script

Prof. Dr. Marcelo Bianchi

*Israel Dragone, Jamison Assunção
Leonardo Fabricius e Rafael Monteiro*

26 de julho de 2018



INSTITUTO DE ASTRONOMIA,
GEOFÍSICA E CIÊNCIAS
ATMOSFÉRICAS

Operadores aritméticos



Operadores aritméticos

- ▶ **Adição:** $\$a+1$, $\$a+\b , $1+2$

Operadores aritméticos

- ▶ **Adição:** $a+1$, $a+b$, $1+2$
- ▶ **Subtração:** $a-1$, $a-b$, $1-2$

Operadores aritméticos

- ▶ **Adição:** $a+1$, $a+b$, $1+2$
- ▶ **Subtração:** $a-1$, $a-b$, $1-2$
- ▶ **Multiplicação:** $a*1$, $a*b$, $1*2$

Operadores aritméticos

- ▶ **Adição:** $a+1$, $a+b$, $1+2$
- ▶ **Subtração:** $a-1$, $a-b$, $1-2$
- ▶ **Multiplicação:** $a*1$, $a*b$, $1*2$
- ▶ **Divisão:** $a/1$, a/b , $1/2$

Operadores aritméticos

- ▶ **Adição:** $a+1$, $a+b$, $1+2$
- ▶ **Subtração:** $a-1$, $a-b$, $1-2$
- ▶ **Multiplicação:** $a*1$, $a*b$, $1*2$
- ▶ **Divisão:** $a/1$, a/b , $1/2$
- ▶ **Módulo:** $a\%1$, $a\%b$, $1\%2$

Operadores aritméticos

- ▶ **Adição:** $\$a+1$, $\$a+\b , $1+2$
- ▶ **Subtração:** $\$a-1$, $\$a-\b , $1-2$
- ▶ **Multiplicação:** $\$a*1$, $\$a*\b , $1*2$
- ▶ **Divisão:** $\$a/1$, $\$a/\b , $1/2$
- ▶ **Módulo:** $\$a\%1$, $\$a\%\b , $1\%2$
- ▶ **Exponenciação:** $\$a**1$, $\$a**\b , $1**2$

Operadores aritméticos

- ▶ **Adição:** $a+1$, $a+b$, $1+2$
- ▶ **Subtração:** $a-1$, $a-b$, $1-2$
- ▶ **Multiplicação:** $a*1$, $a*b$, $1*2$
- ▶ **Divisão:** $a/1$, a/b , $1/2$
- ▶ **Módulo:** $a\%1$, $a\%b$, $1\%2$
- ▶ **Exponenciação:** $a**1$, $a**b$, $1**2$
- ▶ **Incremento:** **let** $i++$, **let** $i+=1$

Operadores aritméticos

- ▶ **Adição:** $\$a+1$, $\$a+\b , $1+2$
- ▶ **Subtração:** $\$a-1$, $\$a-\b , $1-2$
- ▶ **Multiplicação:** $\$a*1$, $\$a*\b , $1*2$
- ▶ **Divisão:** $\$a/1$, $\$a/\b , $1/2$
- ▶ **Módulo:** $\$a\%1$, $\$a\%\b , $1\%2$
- ▶ **Exponenciação:** $\$a**1$, $\$a**\b , $1**2$
- ▶ **Incremento:** **let** $i++$, **let** $i+=1$

Observações

Operadores aritméticos

- ▶ **Adição:** $\$a+1$, $\$a+\b , $1+2$
- ▶ **Subtração:** $\$a-1$, $\$a-\b , $1-2$
- ▶ **Multiplicação:** $\$a*1$, $\$a*\b , $1*2$
- ▶ **Divisão:** $\$a/1$, $\$a/\b , $1/2$
- ▶ **Módulo:** $\$a\%1$, $\$a\%\b , $1\%2$
- ▶ **Exponenciação:** $\$a**1$, $\$a**\b , $1**2$
- ▶ **Incremento:** **let** $i++$, **let** $i+=1$

Observações

- ▶ $\$ soma = \$((a+b))$

Operadores aritméticos

- ▶ **Adição:** $\$a+1$, $\$a+\b , $1+2$
- ▶ **Subtração:** $\$a-1$, $\$a-\b , $1-2$
- ▶ **Multiplicação:** $\$a*1$, $\$a*\b , $1*2$
- ▶ **Divisão:** $\$a/1$, $\$a/\b , $1/2$
- ▶ **Módulo:** $\$a\%1$, $\$a\%\b , $1\%2$
- ▶ **Exponenciação:** $\$a**1$, $\$a**\b , $1**2$
- ▶ **Incremento:** **let** $i++$, **let** $i+=1$

Observações

- ▶ $\$ soma = \$((a+b))$
- ▶ $\$ soma = \$[a+b]$

Operadores aritméticos

- ▶ **Adição:** $\$a+1$, $\$a+\b , $1+2$
- ▶ **Subtração:** $\$a-1$, $\$a-\b , $1-2$
- ▶ **Multiplicação:** $\$a*1$, $\$a*\b , $1*2$
- ▶ **Divisão:** $\$a/1$, $\$a/\b , $1/2$
- ▶ **Módulo:** $\$a\%1$, $\$a\%\b , $1\%2$
- ▶ **Exponenciação:** $\$a**1$, $\$a**\b , $1**2$
- ▶ **Incremento:** **let** $i++$, **let** $i+=1$

Observações

- ▶ $\$ soma = \$((a+b))$
- ▶ $\$ soma = \$[a+b]$

Operadores aritméticos

- ▶ **Adição:** $\$a+1$, $\$a+\b , $1+2$
- ▶ **Subtração:** $\$a-1$, $\$a-\b , $1-2$
- ▶ **Multiplicação:** $\$a*1$, $\$a*\b , $1*2$
- ▶ **Divisão:** $\$a/1$, $\$a/\b , $1/2$
- ▶ **Módulo:** $\$a\%1$, $\$a\%\b , $1\%2$
- ▶ **Exponenciação:** $\$a**1$, $\$a**\b , $1**2$
- ▶ **Incremento:** **let** $i++$, **let** $i+=1$

Observações

- ▶ $\$ soma = \$((a+b))$
- ▶ $\$ soma = \$[a+b]$
- ▶ $\$ \text{echo } \$((a+b))$

Operadores aritméticos

- ▶ **Adição:** $\$a+1$, $\$a+\b , $1+2$
- ▶ **Subtração:** $\$a-1$, $\$a-\b , $1-2$
- ▶ **Multiplicação:** $\$a*1$, $\$a*\b , $1*2$
- ▶ **Divisão:** $\$a/1$, $\$a/\b , $1/2$
- ▶ **Módulo:** $\$a\%1$, $\$a\%\b , $1\%2$
- ▶ **Exponenciação:** $\$a**1$, $\$a**\b , $1**2$
- ▶ **Incremento:** **let** $i++$, **let** $i+=1$

Observações

- ▶ $\$ soma = \$((a+b))$
- ▶ $\$ soma = \#[a+b]$
- ▶ $\$ \text{echo } \$((a+b))$
- ▶ $\$ \text{echo } \#[a+b]$

Operação com floats



Operação com floats

```
$ echo $[10/3]
```



Operação com floats

```
$ echo ${10/3}
```

```
3
```

Operação com floats

```
$ echo ${10/3}
```

```
3
```

```
$ echo ${10/1.5}
```

Operação com floats

```
$ echo ${10/3}
```

```
3
```

```
$ echo ${10/1.5}
```

```
syntax error: invalid arithmetic operator
```

Operação com floats

```
$ echo $[10/3]
```

```
3
```

```
$ echo $[10/1.5]
```

```
syntax error: invalid arithmetic operator
```

```
$ echo $[2.2*2]
```

Operação com floats

```
$ echo ${10/3}
```

```
3
```

```
$ echo ${10/1.5}
```

```
syntax error: invalid arithmetic operator
```

```
$ echo ${2.2*2}
```

```
syntax error: invalid arithmetic operator
```

Operação com floats

```
$ echo ${10/3}
```

```
3
```

```
$ echo ${10/1.5}
```

```
syntax error: invalid arithmetic operator
```

```
$ echo ${2.2*2}
```

```
syntax error: invalid arithmetic operator
```

```
$ echo ${2.2+3}
```

Operação com floats

```
$ echo ${10/3}
```

```
3
```

```
$ echo ${10/1.5}
```

```
syntax error: invalid arithmetic operator
```

```
$ echo ${2.2*2}
```

```
syntax error: invalid arithmetic operator
```

```
$ echo ${2.2+3}
```

```
syntax error: invalid arithmetic operator
```


Operação com floats

```
$ echo ${10/3}
```

```
3
```

```
$ echo ${10/1.5}
```

```
syntax error: invalid arithmetic operator
```

```
$ echo ${2.2*2}
```

```
syntax error: invalid arithmetic operator
```

```
$ echo ${2.2+3}
```

```
syntax error: invalid arithmetic operator
```

O shell não sabe trabalhar com números decimais!!

Bash calculator (bc)



Bash calculator (bc)

```
$ echo "scale=2; 10/3" | bc
```

```
# 10/3
```



Bash calculator (bc)

```
$ echo "scale=2; 10/3" | bc
```

$\frac{10}{3}$

```
3.33
```

Bash calculator (bc)

```
$ echo "scale=2; 10/3" | bc
```

$\frac{10}{3}$

```
3.33
```

```
$ bc <<< "scale=4; (10/1.5)+3"
```

$(\frac{10}{1.5}) + 3$

Bash calculator (bc)

```
$ echo "scale=2; 10/3" | bc
```

$\frac{10}{3}$

3.33

```
$ bc <<< "scale=4; (10/1.5)+3"
```

$(\frac{10}{1.5}) + 3$

9.6666

Bash calculator (bc)

```
$ echo "scale=2; 10/3" | bc
```

$\frac{10}{3}$

```
3.33
```

```
$ bc <<< "scale=4; (10/1.5)+3"
```

$(\frac{10}{1.5}) + 3$

```
9.6666
```

```
$ bc <<< "scale=2; sqrt(96)"
```

$(\sqrt{96})$

Bash calculator (bc)

```
$ echo "scale=2; 10/3" | bc
```

$\frac{10}{3}$

3.33

```
$ bc <<< "scale=4; (10/1.5)+3"
```

$(\frac{10}{1.5}) + 3$

9.6666

```
$ bc <<< "scale=2; sqrt(96)"
```

$(\sqrt{96})$

9.79

Bash calculator (bc)

```
$ echo "scale=2; 10/3" | bc
```

$\frac{10}{3}$

3.33

```
$ bc <<< "scale=4; (10/1.5)+3"
```

$(\frac{10}{1.5}) + 3$

9.6666

```
$ bc <<< "scale=2; sqrt(96)"
```

$(\sqrt{96})$

9.79

```
$ bc -l <<< "scale=3; s(3.14/4)"
```

$\text{sen}(\frac{\pi}{4})$

Bash calculator (bc)

```
$ echo "scale=2; 10/3" | bc
```

$\frac{10}{3}$

3.33

```
$ bc <<< "scale=4; (10/1.5)+3"
```

$(\frac{10}{1.5}) + 3$

9.6666

```
$ bc <<< "scale=2; sqrt(96)"
```

$(\sqrt{96})$

9.79

```
$ bc -l <<< "scale=3; s(3.14/4)"
```

$\text{sen}(\frac{\pi}{4})$

0.706

Bash calculator (bc)

```
$ echo "scale=2; 10/3" | bc
```

$\frac{10}{3}$

3.33

```
$ bc <<< "scale=4; (10/1.5)+3"
```

$(\frac{10}{1.5}) + 3$

9.6666

```
$ bc <<< "scale=2; sqrt(96)"
```

$(\sqrt{96})$

9.79

```
$ bc -l <<< "scale=3; s(3.14/4)"
```

$\text{sen}(\frac{\pi}{4})$

0.706

```
$ var=$(echo "scale=2; 10/3" | bc)
```

```
$ var=`bc <<< "scale=4; 10/1.5"`
```

Comando test



Comando test

Comparações Numéricas

Comando test

Comparações Numéricas

- ▶ **Igual:** -eq (EQual)

Comando test

Comparações Numéricas

- ▶ **Igual:** -eq (EQual)
- ▶ **Diferente:** -ne (Not Equal)

Comando test

Comparações Numéricas

- ▶ **Igual:** -eq (EQual)
- ▶ **Diferente:** -ne (Not Equal)
- ▶ **Maior:** -gt (Greater Than)

Comando test

Comparações Numéricas

- ▶ **Igual:** -eq (Equal)
- ▶ **Diferente:** -ne (Not Equal)
- ▶ **Maior:** -gt (Greater Than)
- ▶ **Maior ou igual:** -ge (Greater or Equal)

Comando test

Comparações Numéricas

- ▶ **Igual:** -eq (Equal)
- ▶ **Diferente:** -ne (Not Equal)
- ▶ **Maior:** -gt (Greater Than)
- ▶ **Maior ou igual:** -ge (Greater or Equal)
- ▶ **Menor:** -lt (Less Than)

Comando test

Comparações Numéricas

- ▶ **Igual:** -eq (Equal)
- ▶ **Diferente:** -ne (Not Equal)
- ▶ **Maior:** -gt (Greater Than)
- ▶ **Maior ou igual:** -ge (Greater or Equal)
- ▶ **Menor:** -lt (Less Than)
- ▶ **Menor ou igual:** -le (Less or Equal)

Comando test

Comparações Numéricas

- ▶ **Igual:** -eq (Equal)
- ▶ **Diferente:** -ne (Not Equal)
- ▶ **Maior:** -gt (Greater Than)
- ▶ **Maior ou igual:** -ge (Greater or Equal)
- ▶ **Menor:** -lt (Less Than)
- ▶ **Menor ou igual:** -le (Less or Equal)
- ▶ **Operadores lógicos:** && (AND), || (OR) ! (Not)

Exemplos

Comando test

Comparações Numéricas

- ▶ **Igual:** -eq (Equal)
- ▶ **Diferente:** -ne (Not Equal)
- ▶ **Maior:** -gt (Greater Than)
- ▶ **Maior ou igual:** -ge (Greater or Equal)
- ▶ **Menor:** -lt (Less Than)
- ▶ **Menor ou igual:** -le (Less or Equal)
- ▶ **Operadores lógicos:** && (AND), || (OR) ! (Not)

Exemplos

- ▶ [\$a -eq \$b], [\$a -eq 10]

Comando test

Comparações Numéricas

- ▶ **Igual:** -eq (Equal)
- ▶ **Diferente:** -ne (Not Equal)
- ▶ **Maior:** -gt (Greater Than)
- ▶ **Maior ou igual:** -ge (Greater or Equal)
- ▶ **Menor:** -lt (Less Than)
- ▶ **Menor ou igual:** -le (Less or Equal)
- ▶ **Operadores lógicos:** && (AND), || (OR) ! (Not)

Exemplos

- ▶ [\$a -eq \$b], [\$a -eq 10]
- ▶ [\$a -gt \$b], [\$a -gt 10]

Comando test

Comparações Numéricas

- ▶ **Igual:** -eq (Equal)
- ▶ **Diferente:** -ne (Not Equal)
- ▶ **Maior:** -gt (Greater Than)
- ▶ **Maior ou igual:** -ge (Greater or Equal)
- ▶ **Menor:** -lt (Less Than)
- ▶ **Menor ou igual:** -le (Less or Equal)
- ▶ **Operadores lógicos:** && (AND), || (OR) ! (Not)

Exemplos

- ▶ [\$a -eq \$b], [\$a -eq 10]
- ▶ [\$a -gt \$b], [\$a -gt 10]
- ▶ [\$a -le \$b] && [\$a -gt 10]

Comando test

Comparações Numéricas

- ▶ **Igual:** -eq (Equal)
- ▶ **Diferente:** -ne (Not Equal)
- ▶ **Maior:** -gt (Greater Than)
- ▶ **Maior ou igual:** -ge (Greater or Equal)
- ▶ **Menor:** -lt (Less Than)
- ▶ **Menor ou igual:** -le (Less or Equal)
- ▶ **Operadores lógicos:** && (AND), || (OR) ! (Not)

Exemplos

- ▶ [\$a -eq \$b], [\$a -eq 10]
- ▶ [\$a -gt \$b], [\$a -gt 10]
- ▶ [\$a -le \$b] && [\$a -gt 10]
- ▶ [[(\$a -gt 3 && \$a -lt 10) || (\$b -eq 3)]]

Comando test



Comando test

Comparação de Strings

- **Igual:** "\$a" = "\$b", "\$a" = "linux"

Comando test

Comparação de Strings

- **Igual:** "\$a" = "\$b", "\$a" = "linux"

Comando test

Comparação de Strings

- ▶ **Igual:** "\$a" = "\$b", "\$a" = "linux"
- ▶ **Diferente:** "\$a" != "\$b", "\$a" != "curso linux"

Comando test

Comparação de Strings

- ▶ **Igual:** "\$a" = "\$b", "\$a" = "linux"
- ▶ **Diferente:** "\$a" != "\$b", "\$a" != "curso linux"
- ▶ **Não Nula:** [-n "\$a"]

Comando test

Comparação de Strings

- ▶ **Igual:** "\$a" = "\$b", "\$a" = "linux"
- ▶ **Diferente:** "\$a" != "\$b", "\$a" != "curso linux"
- ▶ **Não Nula:** [-n "\$a"]
- ▶ **Nula:** [-z "\$a"]

Comando test

Comparação de Strings

- ▶ **Igual:** `"$a" = "$b"`, `"$a" = "linux"`
- ▶ **Diferente:** `"$a" != "$b"`, `"$a" != "curso linux"`
- ▶ **Não Nula:** `[-n "$a"]`
- ▶ **Nula:** `[-z "$a"]`

Teste com arquivos

Comando test

Comparação de Strings

- ▶ **Igual:** "\$a" = "\$b", "\$a" = "linux"
- ▶ **Diferente:** "\$a" != "\$b", "\$a" != "curso linux"
- ▶ **Não Nula:** [-n "\$a"]
- ▶ **Nula:** [-z "\$a"]

Teste com arquivos

- ▶ **Arquivo existe:** [-e "\$filename"]

Comando test

Comparação de Strings

- ▶ **Igual:** "\$a" = "\$b", "\$a" = "linux"
- ▶ **Diferente:** "\$a" != "\$b", "\$a" != "curso linux"
- ▶ **Não Nula:** [-n "\$a"]
- ▶ **Nula:** [-z "\$a"]

Teste com arquivos

- ▶ **Arquivo existe:** [-e "\$filename"]
- ▶ **Permissão de leitura:** [-r "\$filename"]

Comando test

Comparação de Strings

- ▶ **Igual:** "\$a" = "\$b", "\$a" = "linux"
- ▶ **Diferente:** "\$a" != "\$b", "\$a" != "curso linux"
- ▶ **Não Nula:** [-n "\$a"]
- ▶ **Nula:** [-z "\$a"]

Teste com arquivos

- ▶ **Arquivo existe:** [-e "\$filename"]
- ▶ **Permissão de leitura:** [-r "\$filename"]
- ▶ **Permissão de escrita:** [-w "\$filename"]

Comando test

Comparação de Strings

- ▶ **Igual:** "\$a" = "\$b", "\$a" = "linux"
- ▶ **Diferente:** "\$a" != "\$b", "\$a" != "curso linux"
- ▶ **Não Nula:** [-n "\$a"]
- ▶ **Nula:** [-z "\$a"]

Teste com arquivos

- ▶ **Arquivo existe:** [-e "\$filename"]
- ▶ **Permissão de leitura:** [-r "\$filename"]
- ▶ **Permissão de escrita:** [-w "\$filename"]
- ▶ **Permissão de execução:** [-x "\$filename"]

Como usar essas comparações para estruturar ações?



Como usar essas comparações para estruturar ações?

```
if [ condição ]; then  
    ações  
fi
```



Como usar essas comparações para estruturar ações?

```
if [ condição ]; then  
    ações  
fi
```

```
cor="azul"
```

Como usar essas comparações para estruturar ações?

```
if [ condição ]; then  
    ações  
fi
```

```
cor="azul"
```

```
if [ "$cor" = "azul" ]; then  
    echo "A cor escolhida foi azul"  
fi
```

```
cor="vermelho"
```

Como usar essas comparações para estruturar ações?

```
if [ condição ]; then
```

```
    ações
```

```
fi
```

```
cor="azul"
```

```
if [ "$cor" = "azul" ]; then
```

```
    echo "A cor escolhida foi azul"
```

```
fi
```

```
cor="vermelho"
```

```
if [ "$cor" = "azul" ]; then
```

```
    echo "A cor escolhida foi azul"
```

```
else
```

```
    echo "A cor escolhida não foi azul"
```

```
fi
```


Estruturando condições



Estruturando condições

```
if [ "$cor" = "azul" ]; then
    echo "A cor escolhida foi azul"

elif [ "$cor" = "verde" ]; then
    echo "A cor escolhida foi verde"

elif [ "$cor" = "rosa" ]; then
    echo "A cor escolhida foi rosa"

elif [ "$cor" = "vermelho" ]; then
    echo "A cor escolhida foi vermelho"

else
    echo "Cor inválida"

fi
```

Testando várias opções - case



Testando várias opções - case

case "\$cor" in

azul echo "A cor escolhida foi azul" ;;

verde echo "A cor escolhida foi verde" ;;

rosa echo "A cor escolhida foi rosa" ;;

vermelho echo "A cor escolhida foi vermelho" ;;

***) echo** "Opção inválida" ;;

esac

Exercícios



Exercícios

1. Faça um script que receba um número inteiro do teclado (n) e exiba tela: (i) o número lido (n); (ii) $n+5$; (iii) se n é par ou ímpar e (iv) n^2 .
2. Faça um script que receba dois números inteiros do teclado (n_1 e n_2) e exiba na tela: (i) qual foi o número maior lido; (ii) se n_1 e n_2 são pares ou ímpares ou se um é par e outro é ímpar; (iii) calcule a raiz quadrada do menor número e (iv) e^{n_2} .
3. Faça um script que peça ao usuário que digite o nome de um dia da semana e dê como saída quantos dias faltam para tal dia e qual dia do mês será. (Ex. se o dia inserido for sábado a saída deverá ser: o próximo sábado será em 2 dias, dia 28 do mês).
4. Verifique quais permissões o script do exercício 2 possui e exiba na tela a soma delas. (lembre-se: $r=4$, $w=2$ e $x=1$).

Laços de repetição - lista



Laços de repetição - lista

```
for i in abacate 1 e camelo 2 gnu  
do  
    echo -n "$i "  
done
```



Laços de repetição - lista

```
for i in abacate 1 e camelo 2 gnu  
do
```

```
    echo -n "$i "
```

```
done
```

```
abacate 1 e camelo 2 gnu
```



Laços de repetição - lista

```
for i in abacate 1 e camelo 2 gnu
do
    echo -n "$i "
done
```

```
abacate 1 e camelo 2 gnu
```

```
for i in /cursoLinux/exercicios/*
do
    echo -n "$i "
done
```

Laços de repetição - lista

```
for i in abacate 1 e camelo 2 gnu  
do  
    echo -n "$i "  
done
```

```
abacate 1 e camelo 2 gnu
```

```
for i in /cursoLinux/exercicios/*  
do  
    echo -n "$i "  
done
```

```
aula1 aula2 aula3 aula4 aula5 aula6 aula7 aula8 aula9
```

Laços de repetição - lista

```
for i in abacate 1 e camelo 2 gnu
do
    echo -n "$i "
done
```

```
abacate 1 e camelo 2 gnu
```

```
for i in /cursoLinux/exercicios/*
do
    echo -n "$i "
done
```

```
aula1 aula2 aula3 aula4 aula5 aula6 aula7 aula8 aula9
```

```
for i in `awk 'NR>1 && NR <6 print $1' premier_league.dat`
do
    echo -n "$i "
done
```

Laços de repetição - lista

```
for i in abacate 1 e camelo 2 gnu  
do
```

```
    echo -n "$i "
```

```
done
```

```
abacate 1 e camelo 2 gnu
```

```
for i in /cursoLinux/exercicios/*  
do
```

```
    echo -n "$i "
```

```
done
```

```
aula1 aula2 aula3 aula4 aula5 aula6 aula7 aula8 aula9
```

```
for i in `awk 'NR>1 && NR <6 print $1' premier_league.dat`  
do
```

```
    echo -n "$i "
```

```
done
```

```
Manchester_City Manchester_United Tottenham Liverpool
```

Laços de repetição (seq)



Laços de repetição (seq)

```
for i in $(seq 1 10)
do
    echo -n "$i "
done
```



Laços de repetição (seq)

```
for i in $(seq 1 10)
do
    echo -n "$i "
done
```

```
1 2 3 4 5 6 7 8 9 10
```


Laços de repetição (seq)

```
for i in $(seq 1 10)
do
    echo -n "$i "
done
```

```
1 2 3 4 5 6 7 8 9 10
```

```
for i in $(seq 1 2 10)
do
    echo -n "$i "
done
```

Laços de repetição (seq)

```
for i in $(seq 1 10)
do
    echo -n "$i "
done
```

```
1 2 3 4 5 6 7 8 9 10
```

```
for i in $(seq 1 2 10)
do
    echo -n "$i "
done
```

```
1 3 5 7 9
```

Laços de repetição (seq)

```
for i in $(seq 1 10)
do
    echo -n "$i "
done
```

```
1 2 3 4 5 6 7 8 9 10
```

```
for i in $(seq 1 2 10)
do
    echo -n "$i "
done
```

```
1 3 5 7 9
```

```
for i in $(seq 1 -1 -10)
do
    echo -n "$i "
done
```

Laços de repetição (seq)

```
for i in $(seq 1 10)
do
    echo -n "$i "
done
```

```
1 2 3 4 5 6 7 8 9 10
```

```
for i in $(seq 1 2 10)
do
    echo -n "$i "
done
```

```
1 3 5 7 9
```

```
for i in $(seq 1 -1 -10)
do
    echo -n "$i "
done
```

```
1 0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10
```

Laços de repetição - {}



Laços de repetição - {}

```
for i in {1..10}  
do  
    echo -n "$i "  
done
```



Laços de repetição - {}

```
for i in {1..10}
do
    echo -n "$i "
done
```

```
1 2 3 4 5 6 7 8 9 10
```

Laços de repetição - {}

```
for i in {1..10}
do
    echo -n "$i "
done
```

```
1 2 3 4 5 6 7 8 9 10
```

```
for i in {1..10..2}
do
    echo -n "$i "
done
```


Laços de repetição - {}

```
for i in {1..10}
do
    echo -n "$i "
done
```

```
1 2 3 4 5 6 7 8 9 10
```

```
for i in {1..10..2}
do
    echo -n "$i "
done
```

```
1 3 5 7 9
```

Laços de repetição - {}

```
for i in {1..10}
do
    echo -n "$i "
done
```

```
1 2 3 4 5 6 7 8 9 10
```

```
for i in {1..10..2}
do
    echo -n "$i "
done
```

```
1 3 5 7 9
```

```
for i in {A..Z}
do
    echo -n "$i "
done
```

Laços de repetição - {}

```
for i in {1..10}
do
    echo -n "$i "
done
```

```
1 2 3 4 5 6 7 8 9 10
```

```
for i in {1..10..2}
do
    echo -n "$i "
done
```

```
1 3 5 7 9
```

```
for i in {A..Z}
do
    echo -n "$i "
done
```

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

Laços de repetição - padrão C



Laços de repetição - padrão C

```
for ((i=1;i<10;i++))  
do  
    echo -n "$i "  
done
```



Laços de repetição - padrão C

```
for ((i=1;i<10;i++))  
do  
    echo -n "$i "  
done
```

```
1 2 3 4 5 6 7 8 9 10
```

Laços de repetição - padrão C

```
for ((i=1;i<10;i++))  
do  
    echo -n "$i "  
done
```

```
1 2 3 4 5 6 7 8 9 10
```

```
for ((i=1;i<10;i+=2))  
do  
    echo -n "$i "  
done
```

Laços de repetição - padrão C

```
for ((i=1;i<10;i++))  
do  
    echo -n "$i "  
done
```

```
1 2 3 4 5 6 7 8 9 10
```

```
for ((i=1;i<10;i+=2))  
do  
    echo -n "$i "  
done
```

```
1 3 5 7 9
```


Laços de repetição - padrão C

```
for ((i=1;i<10;i++))  
do  
    echo -n "$i "  
done
```

```
1 2 3 4 5 6 7 8 9 10
```

```
for ((i=1;i<10;i+=2))  
do  
    echo -n "$i "  
done
```

```
1 3 5 7 9
```

```
for ((i=1;i>-10;i--))  
do  
    echo -n "$i "  
done
```

Laços de repetição - padrão C

```
for ((i=1;i<10;i++))  
do  
    echo -n "$i "  
done
```

```
1 2 3 4 5 6 7 8 9 10
```

```
for ((i=1;i<10;i+=2))  
do  
    echo -n "$i "  
done
```

```
1 3 5 7 9
```

```
for ((i=1;i>-10;i--))  
do  
    echo -n "$i "  
done
```

```
1 0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10
```

Laços de repetição - while



Laços de repetição - while

```
while [ $var -ne 0 ];  
do  
    ações  
done
```

Laços de repetição - while

```
while [ $var -ne 0 ];
```

```
do
```

```
    ações
```

```
done
```

```
var="s"
```

```
while [ "$var" != "n" ];
```

```
do
```

```
    read -p "Digite um número para saber seu quadrado" num
```

```
    echo "$num ao quadrado é ${num**2}"
```

```
    read -p "Deseja entrar com outro número? (s/n)" var
```

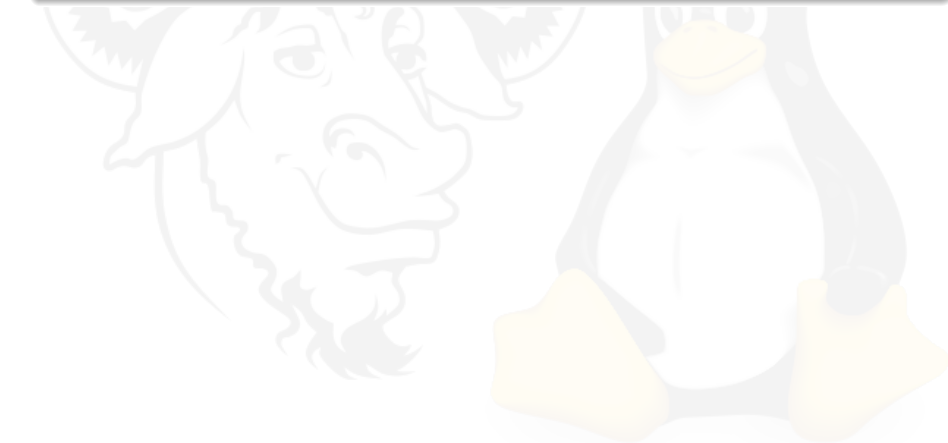
```
done
```

Funções



Funções

```
function nomeFuncao( ) {  
    código  
}
```



Funções

```
function nomeFuncao( ) {  
    código  
}
```

```
function fatorial( ) {  
    if [ "$1" -gt "1" ]; then  
        i=${1-1}  
        j=`fatorial $i`  
        k=${1*$j}  
        echo $k  
    else  
        echo 1  
    fi  
}
```


Funções

```
function nomeFuncao( ) {  
    código  
}
```

```
function fatorial( ) {  
    if [ "$1" -gt "1" ]; then  
        i=${1-1}  
        j=`fatorial $i`  
        k=${1*$j}  
        echo $k  
    else  
        echo 1  
    fi  
}
```

fatorial 5

Funções



Funções

```
function operacoes( ) {  
    n1=$1 n2=$2  
    echo "n1*n2 = ${n1*n2}"  
    echo "n1^n2 = ${n1**n2}"  
    echo `bc -l <<< "scale=2; sqrt(n1) * l(n2)"`  
}
```

Funções

```
function operacoes( ) {  
    n1=$1 n2=$2  
    echo "n1*n2 = ${n1*n2}"  
    echo "n1^n2 = ${n1**n2}"  
    echo `bc -l <<< "scale=2; sqrt(n1) * l(n2)"`  
}
```

operacoes 8 3

Funções

```
function operacoes( ) {  
    n1=$1 n2=$2  
    echo "n1*n2 = ${n1*n2}"  
    echo "n1^n2 = ${n1**n2}"  
    echo `bc -l <<< "scale=2; sqrt(n1) * l(n2)"`  
}
```

operacoes 8 3

24
512
3.07

Funções

```
function operacoes( ) {  
    n1=$1 n2=$2  
    echo "n1*n2 = ${n1*n2}"  
    echo "n1^n2 = ${n1**n2}"  
    echo `bc -l <<< "scale=2; sqrt(n1) * l(n2)"`  
}
```

operacoes 8 3

24
512
3.07

- ▶ **\$0**: Armazena o nome da função

Funções

```
function operacoes( ) {  
    n1=$1 n2=$2  
    echo "n1*n2 = ${n1*n2}"  
    echo "n1^n2 = ${n1**n2}"  
    echo `bc -l <<< "scale=2; sqrt(n1) * l(n2)"`  
}
```

operacoes 8 3

24
512
3.07

- ▶ **\$0**: Armazena o nome da função
- ▶ **\$1 ... \$n**: Armazena os argumentos de 1 a n passados a função

Funções

```
function operacoes( ) {  
    n1=$1 n2=$2  
    echo "n1*n2 = ${n1*n2}"  
    echo "n1^n2 = ${n1**n2}"  
    echo `bc -l <<< "scale=2; sqrt(n1) * l(n2)"`  
}
```

operacoes 8 3

24
512
3.07

- ▶ **\$0**: Armazena o nome da função
- ▶ **\$1 ... \$n**: Armazena os argumentos de 1 a n passados a função
- ▶ **\$#**: Armazena o número de argumentos passados à função

Funções

```
function operacoes( ) {  
    n1=$1 n2=$2  
    echo "n1*n2 = ${n1*n2}"  
    echo "n1^n2 = ${n1**n2}"  
    echo `bc -l <<< "scale=2; sqrt(n1) * l(n2)"`  
}
```

operacoes 8 3

24
512
3.07

- ▶ **\$0**: Armazena o nome da função
- ▶ **\$1 ... \$n**: Armazena os argumentos de 1 a n passados a função
- ▶ **\$#**: Armazena o número de argumentos passados à função
- ▶ **\$***: Exibe todos os argumentos passados à função

Exercícios



Exercícios

5. Crie um script que dado um número inteiro calcula sua tabuada de 1 até 20. Salve a saída num arquivo de texto chamado `tabuada_do_$num.dat`.
6. Sabendo que a letra A ocupa a posição 1 no alfabeto, B a 2, C a 3 ... Faça um código que exiba as letras em posições pares ou ímpares de acordo com a escolha do usuário.
7. Faça um código que receba dois números do teclado e uma operação (soma, subtração, multiplicação e divisão). Para cada operação crie uma função distinta que deve ser chamada de acordo com a escolha feita pelo usuário. Mostre na tela os números dados, a operação realizada e seu resultado.
8. Crie um script que recebe um número inteiro e exiba na tela um triângulo retângulo de lado igual ao número fornecido utilizando o caracter *, por exemplo. Após exibir o triângulo pergunte ao usuário se ele quer entrar com outro número ou encerrar a execução.

Exercícios



Exercícios

9. Repita o exercício do convert da aula IV de conversão da imagem para jpg. Agora salve uma imagem para cada nível de compressão (1 a 100 variando de 1 em 1), salve esses arquivos dentro de uma pasta (Dica, para visualizar uma imagem jpg pelo terminal execute `$ eog imagem.jpg`).
10. Utilizando o catálogo de terremotos da IRIS da aula V faça uma lista do número de terremotos por intervalo de magnitude entre: 1 e 2, 2 e 3 ... 8 e 9 e maiores que 9. (Na busca procure por terremotos $\geq x$ e $< x+1$)
11. Ainda utilizando o catálogo da IRIS faça uma lista de todos os eventos ocorridos no Brasil em ordem decrescente de magnitude, salvando no arquivo: longitude, latitude, data e magnitude.